FMT_matching_analysis.Rmd

Code

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com (http://rmarkdown.rstudio.com).

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

Hide

```
setwd(dirname(rstudioapi::getActiveDocumentContext()$path))
getwd()
```

```
[1] "C:/work/fmt_enterotype/a_microbiome/analysis"
```

Hide

```
source('./pre_processing.R')
```

```
[1] 0.01
[1] 0.01
[1] 0.01
[1] 0.01
```

Hide

```
meta_fil_config1_na <- meta_fil_config1[!meta_fil_config1$postfmt_symptoms %in% c(NA),]

table(meta_fil_config1_na$postfmt_symptoms, ifelse(meta_fil_config1_na$Diease1 == 'CDI', 'CDI',
'IBD'), meta_fil_config1_na$don_entro, meta_fil_config1_na$pre_entro)
```

```
, ,  = donor1,  = before1


          CDI IBD
  failure   22  16
  response  90   8

, ,  = donor2,  = before1


          CDI IBD
  failure    9   6
  response  23   8

, ,  = donor1,  = before2


          CDI IBD
  failure    8  15
  response  28  24

, ,  = donor2,  = before2


          CDI IBD
  failure    3  16
  response   5   5
```

Hide

```
meta_config_in_this <- meta_fil_config1_na
```

Hide

```
#alpha
tmp_arare <- arare
rownames(tmp_arare) <- tmp_arare$X
# tmp_arare <- tmp_arare[rownames(before_donor_L6), c('X', 'shannon')]
cbind_arare <- cbind(tmp_arare[meta_config_in_this$Donor_sra, c('X', 'shannon')], tmp_arare[meta_config_in_this$Previous_sra, c('X', 'shannon')])#
if(unique(cbind_arare[,1] == meta_config_in_this$Donor_sra & cbind_arare[,3] == meta_config_in_this$Previous_sra)==c(TRUE)){
    cn_arare = cbind(meta_config_in_this$SRA_Sample, cbind_arare, meta_config_in_this$postfmt_symptoms, meta_config_in_this$pre_entro, meta_config_in_this$don_entro, meta_config_in_this$PRJ, meta_config_in_this$Diease1)
    colnames(cn_arare) <- c('names', 'Dnames', 'Darare', 'Pnames', 'Parare', 'symptom', 'pre_entro', 'don_entro', 'PRJ', 'Diease1')
}
```

Hide

```
DP_distance <- NULL
len <- nrow(meta_config_in_this)
L6_rela_fil_sAg_others_remove <- noise.removal(L6_rela_fil_sAg_others, percent=0.01)
```

```
[1] 0.01
```

Hide

```
for(i in 1:len){
    x1 <- as.numeric(as.character(L6_rela_fil_sAg_others_remove[, meta_config_in_this[i, "Donor_sra"]]))
    x2 <- as.numeric(as.character(L6_rela_fil_sAg_others_remove[, meta_config_in_this[i, "Previous_sra"]]))
    x3 <- as.numeric(as.character(L6_rela_fil_sAg_others_remove[, meta_config_in_this[i, "SRA_Sample"]]))
    DP_distance <- rbind(DP_distance, c(meta_config_in_this[i, "SRA_Sample"], vegdist(rbind(x1, x2), method='bray'), vegdist(rbind(x1, x3), method='bray'), vegdist(rbind(x2, x3), method='bray')))
}

DP_distance_dat <- as.data.frame(DP_distance, stringsAsFactors = F)
colnames(DP_distance_dat) <- c('SRA_Sample', 'distance', 'DA_distance', 'PA_distance')
```

Hide

```
if(unique(DP_distance_dat$SRA_Sample == (cn_arare$names))==c(TRUE)){
   DP_distance_dat_arare_div <- cbind(DP_distance_dat, cn_arare)
}

DP_distance_dat_arare_div$distance <- as.numeric(DP_distance_dat_arare_div$distance)
DP_distance_dat_arare_div$DA_distance <- as.numeric(DP_distance_dat_arare_div$DA_distance)
DP_distance_dat_arare_div$PA_distance <- as.numeric(DP_distance_dat_arare_div$PA_distance)
```

Hide

```
DP_distance_dat_arare_div_b1 <- DP_distance_dat_arare_div[DP_distance_dat_arare_div$pre_entro %in% c('before1'),]
DP_distance_dat_arare_div_b2 <- DP_distance_dat_arare_div[DP_distance_dat_arare_div$pre_entro %in% c('before2'),]
```

Hide

```
test_pval<-function(test, symptoms, PRJ, dataframe){
    fc_coin <- as.data.frame(cbind(as.numeric(test), (dataframe[,symptoms]), (dataframe[,PR
J])))
    colnames(fc_coin)<- c('fc', 'group', 'prj')

    prj_list <- fc_coin$prj
    list <- NULL
    for(i in unique(prj_list)){
        if (length(prj_list[prj_list %in% i]) > 2){
            list <- c(list, i)
        }
    }
    fc_coin_f <- fc_coin[fc_coin$prj %in% list,]
    fc_coin_f$prj <- as.factor(as.character(fc_coin_f$prj))
    fc_coin_f$group <- as.factor(as.character(fc_coin_f$group))

    pval<-wilcox_test(fc ~ group | prj, fc_coin_f)
    pval
}
```

Hide

```
##PA distance ~ response
test_pval(DP_distance_dat_arare_div$PA_distance, 'symptom', 'PRJ', DP_distance_dat_arare_div)
```

```

    Asymptotic Wilcoxon-Mann-Whitney Test

data:  fc by group (1, 2)
     stratified by prj
Z = -2.0571, p-value = 0.03968
alternative hypothesis: true mu is not equal to 0
```

Hide

```
test_pval(DP_distance_dat_arare_div[DP_distance_dat_arare_div$pre_entro %in% c('before1'),]$PA_
distance, 'symptom', 'PRJ', DP_distance_dat_arare_div[DP_distance_dat_arare_div$pre_entro %in%
 c('before1'),])
```

```

    Asymptotic Wilcoxon-Mann-Whitney Test

data:  fc by group (1, 2)
     stratified by prj
Z = -2.3808, p-value = 0.01727
alternative hypothesis: true mu is not equal to 0
```

Hide

```
test_pval(DP_distance_dat_arare_div[DP_distance_dat_arare_div$pre_entro %in% c('before2'),]$PA_
distance, 'symptom', 'PRJ', DP_distance_dat_arare_div[DP_distance_dat_arare_div$pre_entro %in%
 c('before2'),])
```

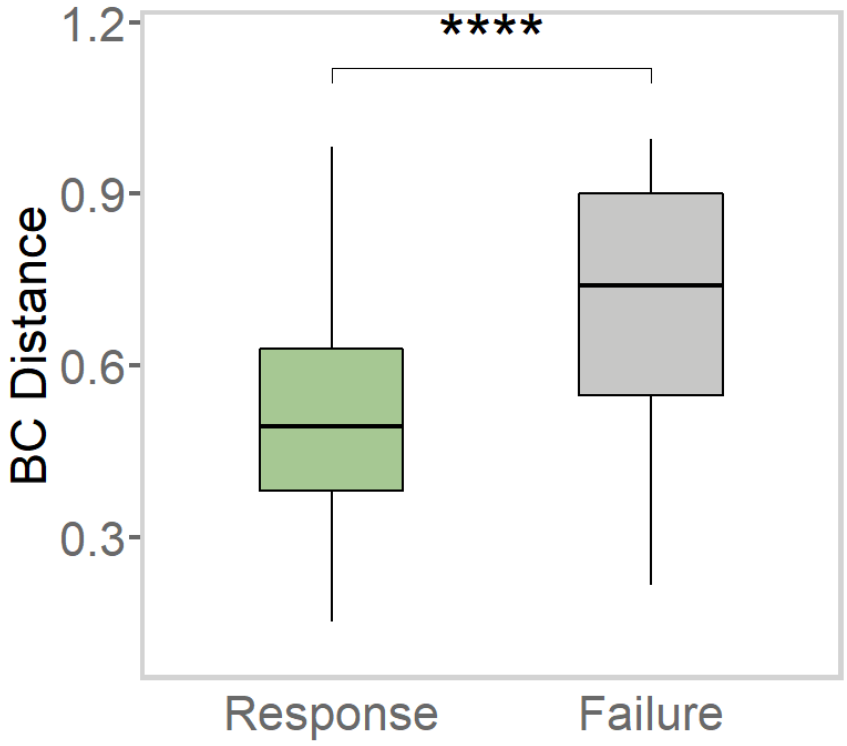
```
    Asymptotic Wilcoxon-Mann-Whitney Test

data:  fc by group (1, 2)
     stratified by prj
Z = -0.92304, p-value = 0.356
alternative hypothesis: true mu is not equal to 0
```

Hide

```
##plot
```

Hide

```
library(ggpubr)
# beeswarm(log(DA_distance+1)~symptom, DP_distance_dat_arare_div)
DP_distance_dat_arare_div_p1 <- DP_distance_dat_arare_div[DP_distance_dat_arare_div$pre_entro %
in% c('before1'),]
DP_distance_dat_arare_div_p1$symptom <- ifelse(DP_distance_dat_arare_div_p1$symptom == 'respons
e', 'Response', 'Failure')
DP_distance_dat_arare_div_p1$symptom <- factor(DP_distance_dat_arare_div_p1$symptom, levels=c(
'Response', 'Failure'))

ggboxplot(DP_distance_dat_arare_div_p1, x = 'symptom', y = ('DA_distance'), fill='symptom', col
or='black', alpha=0.5, size=0.8, width=0.45)+
    scale_y_continuous(expand = c(0, 0.1))+
    stat_compare_means(comparisons = list(c('Response', 'Failure')), method = 'wilcox.test', la
bel = "p.signif", label.y = 1.12, size=16)+
   labs(x= c(''), title=c(''), y = c('BC Distance'))+#Donor to recipient after FMT
 scale_fill_manual(name="FMT", values=c("#4D9127", "#90908D",  donor_before_after_color, "#962E
2B", "#4E86C6", "#4D9127", "#90908D", 'lightgrey'))+#'#C77CFF', '#43AFC8',
  theme(text=element_text(family ="sans", size=32), plot.title = element_text(size=34, hjust =
0.5), axis.text = element_text(size=32, color ='dimgray'), axis.title.x = element_text(size=34
), axis.title.y = element_text(size=34), axis.ticks = element_blank())+
 theme(aspect.ratio = 0.95, legend.background=element_blank(), legend.position=c(1.75, 0.6)
       ,panel.background = element_rect(fill = NA, colour = "lightgrey", size = 3)
       ,axis.line=element_line(colour=NA), axis.ticks.y = element_line(size=1.5, color ='dimgra
y'), axis.ticks.length = unit(7, "pt")
       ,legend.key = element_rect(fill = NA, color = NA))+
   guides(colour = guide_legend(override.aes = list(size=5)))
```



Hide

```
fig4i = 0
ggsave(paste("./figure4/4main_", fig4i, ".pdf", sep = ''), device = "pdf", useDingbats=FALSE)
```

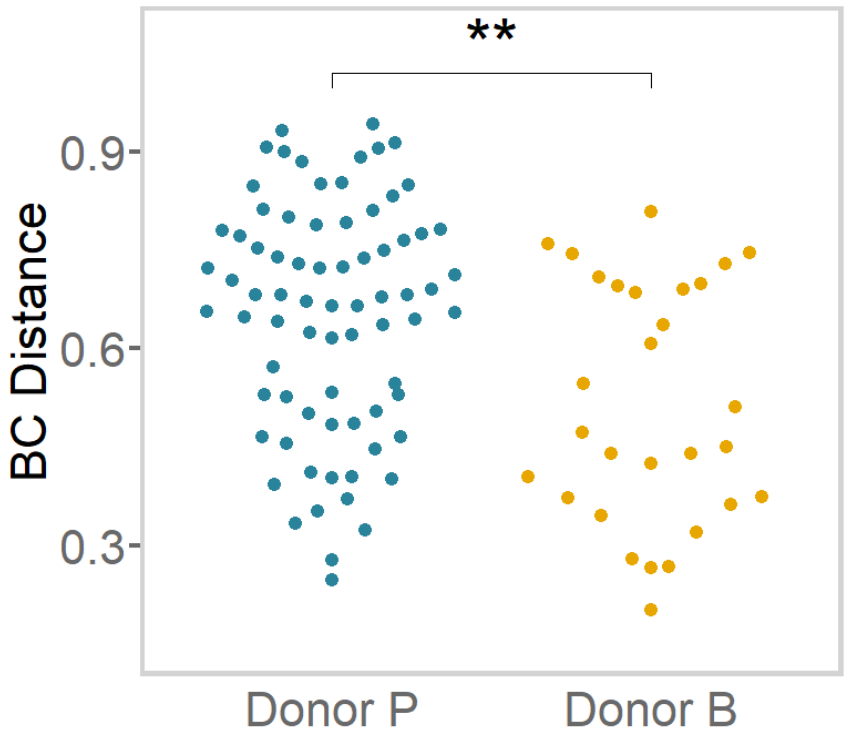
```
Saving 12.9 x 8 in image
```

Hide

```
fig4i = fig4i + 1
```

Hide

```
##distance donor
DP_distance_dat_arare_div_p2 <- DP_distance_dat_arare_div[DP_distance_dat_arare_div$pre_entro %
in% c('before2'),]
DP_distance_dat_arare_div_p2$don_entro <- ifelse(DP_distance_dat_arare_div_p2$don_entro == 'don
or1', 'Donor P', 'Donor B')
DP_distance_dat_arare_div_p2$don_entro <- factor(DP_distance_dat_arare_div_p2$don_entro, levels
=c('Donor P', 'Donor B'))

cex=1.5


library(ggbeeswarm)


ggboxplot(DP_distance_dat_arare_div_p2, x='don_entro', y='distance', alpha = 0, color='white')+
    geom_quasirandom(aes(color=don_entro), method='smiley', size=4)+
    scale_y_continuous(expand = c(0, 0.1))+
    # scale_alpha_continuous( range =  c(0.6, 1))+
  stat_compare_means(comparisons = list(c('Donor P', 'Donor B')), method = 'wilcox.test', label
= "p.signif", label.y = 1.02, size=16)+
  labs(x= c(''), title=c(''),  y= c('BC Distance'))+#Donor to recipient before FMT
    scale_color_manual(name="FMT", values=c('#28839B', '#E7A600', "#4D9127", "#90908D",  donor_
before_after_color, "#962E2B", "#4E86C6", "#4D9127", "#90908D", 'lightgrey'))+
  theme(text=element_text(family ="sans", size=32), plot.title = element_text(size=34, hjust =
0.5), axis.text = element_text(size=32, color ='dimgray'), axis.title.x = element_text(size=34
), axis.title.y = element_text(size=34), axis.ticks = element_blank())+
 theme(aspect.ratio = 0.95, legend.background=element_blank(), legend.position=c(1.75, 0.6)
       ,panel.background = element_rect(fill = NA, colour = "lightgrey", size = 3)
       ,axis.line=element_line(colour=NA), axis.ticks.y = element_line(size=1.5, color ='dimgra
y'), axis.ticks.length = unit(7, "pt")
       ,legend.key = element_rect(fill = NA, color = NA))+
   guides(colour = guide_legend(override.aes = list(size=5)))
```



Hide

```
ggsave(paste("./figure4/4main_distance", fig4i, ".pdf", sep = ''), device = "pdf", useDingbats=FALSE)
```

```
Saving 12.9 x 8 in image
```

Hide

```
fig4i = fig4i + 1
```

Hide

```
DP_distance_dat_arare_div_p2 <- DP_distance_dat_arare_div[DP_distance_dat_arare_div$pre_entro %in% c('before2'),]
DP_distance_dat_arare_div_p2$symptom <- ifelse(DP_distance_dat_arare_div_p2$symptom == 'respons e', 'Response', 'Failure')
DP_distance_dat_arare_div_p2$symptom <- factor(DP_distance_dat_arare_div_p2$symptom, levels=c( 'Response', 'Failure'))

DP_distance_dat_arare_div_p2$c_arare <- DP_distance_dat_arare_div_p2$Darare + DP_distance_dat_a rare_div_p2$Parare
cex=1.5


ggviolin(DP_distance_dat_arare_div_p2, x='symptom', y='c_arare', fill='symptom', alpha = 0.6, w idth = 0.5, color='NA')+
    geom_boxplot(aes(x=symptom, y=c_arare), color='black', alpha=0, width=0.2, DP_distance_dat_ arare_div_p2)+
    scale_y_continuous(expand = c(0, 1))+
    # scale_alpha_continuous( range =  c(0.6, 1))+
    stat_compare_means(comparisons = list(c('Response', 'Failure')), method = 'wilcox.test', la bel = "p.signif",  label.y = 15.8, size=16)+
 scale_fill_manual(name="FMT", values=c("#4D9127", "#90908D",  donor_before_after_color, "#962E 2B", "#4E86C6", "#4D9127", "#90908D", 'lightgrey'))+
  labs(x= c(''), title=c(''),  y= c('Shannon index'))+#Donor + Recipient
  theme(text=element_text(family ="sans", size=32), plot.title = element_text(size=34, hjust = 0.5), axis.text = element_text(size=32, color ='dimgray'), axis.title.x = element_text(size=34 ), axis.title.y = element_text(size=34), axis.ticks = element_blank())+
 theme(aspect.ratio = 0.95, legend.background=element_blank(), legend.position=c(1.75, 0.6)
       ,panel.background = element_rect(fill = NA, colour = "lightgrey", size = 3)
       ,axis.line=element_line(colour=NA), axis.ticks.y = element_line(size=1.5, color ='dimgra y'), axis.ticks.length = unit(7, "pt")
       ,legend.key = element_rect(fill = NA, color = NA))+
   guides(colour = guide_legend(override.aes = list(size=5)))
```

Hide

```
ggsave(paste("./figure4/4main_", fig4i, ".pdf", sep = ''), device = "pdf")
```

```
Saving 12.9 x 8 in image
```

Hide

```
fig4i = fig4i + 1
```

Hide

```
setwd(dirname(rstudioapi::getActiveDocumentContext()$path))
```

Hide

```
cluster_medoids.JSD <- function(inMatrix, pseudocount=0.000001, k=2, ...) {
    KLD <- function(x,y) sum(x *log(x/y))
    JSD<- function(x,y) sqrt(0.5 * KLD(x, (x+y)/2) + 0.5 * KLD(y, (x+y)/2))
    matrixColSize <- length(colnames(inMatrix))
    matrixRowSize <- length(rownames(inMatrix))
    colnames <- colnames(inMatrix)
    # resultsMatrix <- matrix(0, matrixColSize, matrixColSize)
    cluster <- rep(0, matrixColSize)
    inMatrix = apply(inMatrix,1:2,function(x) ifelse (x<=0.0000001,pseudocount,x))

    for(i in 1:(matrixColSize)){
        cluster[i] = ifelse(JSD(as.vector(inMatrix[,i]), as.vector(inMatrix[,matrixColSize])) >
                                JSD(as.vector(inMatrix[,i]), as.vector(inMatrix[,matrixColSize-
1])), 1, 2)
    }
    return(cluster)
}
```

Hide

```
classify_entero <- function(relatives_meta, a, medoids, b, c){
  #'Previous_sra' pre_data.medoids pre_entro 'before'
  relatives_pre_data <- L6_abso[,unique(c(relatives_meta[, a], medoids))]
  sample_size <- colSums(relatives_pre_data)
  relatives_pre_data <- 1 * t(apply(relatives_pre_data, 1, function(x){x/sample_size}))
  relatives_pre_data_remove = noise.removal(relatives_pre_data, percent=0.01)

  relatives_pre_data.cluster= cluster_medoids.JSD(relatives_pre_data_remove, 0.000001, 2)

  samples = dim(relatives_pre_data_remove)[2]
  relatives_meta[, b] <- paste(rep(c, length(1:(samples-2))), relatives_pre_data.cluster[1:(sam
ples-2)], sep = '')
  relatives_meta
}
```

Hide

```
donor_engraft <- function(pre_entro){
    # pre_entro <- "before2"

    tmp_donor_entro <- meta_fil_config1[meta_fil_config1$pre_entro %in% c(pre_entro), c('Donor_
sra', 'don_entro', 'PRJ')]
    tmp_donor_entro_u <- unique(tmp_donor_entro)

    L6_rela_fil_sAg_remove_simp_donor <- L6_rela_fil_sAg_remove_simp[,tmp_donor_entro_u$Donor_s
ra]

    # cols <- ncol(feature_abun_dat)
    group <- as.character(tmp_donor_entro_u$don_entro)
    prj <- tmp_donor_entro_u$PRJ
    seq(0.1, 0.9, 0.05) -> quan

    tmp_pval <- apply(L6_rela_fil_sAg_remove_simp_donor, 1, function(x){
        pt <-  as.data.frame(cbind(x, group, prj))
        colnames(pt)<-c('nx', 'group', 'prj')
        # upt <- unique(pt)
        upt <- pt
        prj_list <- upt$prj
        list <- NULL
        for(i in unique(prj_list)){
            if (length(prj_list[prj_list %in% i]) > 2){
                list <- c(list, i)
            }
        }
        upt <- upt[upt$prj %in% list,]
        upt$nx <- as.numeric(as.character(upt$nx))
        tmp_test <- wilcox_test(nx ~ group | prj, upt)
        # pval_a <- 1
        pval_a <- pvalue(tmp_test)
        if(is.na(pval_a)){pval_a<-1}
        a_nx <- upt[upt$group %in% c('donor1'), 'nx']
        p_nx <- upt[upt$group %in% c('donor2'), 'nx']
        # case <- quantile(log2(a_nx + 0.0001), quan)
        # control <- quantile(log2(p_nx+ 0.0001), quan)
        # gfc <- sum((case - control))/length(quan)

        gfc <- log2(mean(a_nx)/mean(p_nx))

        return(c(pval_a, gfc))
    })

    ###select marked genus qvalue<0.05, and combine abundance
    tmp_pval_t <- data.frame(t(tmp_pval))
    colnames(tmp_pval_t) <- c('pval', 'gfc')
    # tmp_pval_t$id <- rownames(tmp_pval_t)
    # entero_diff_t$pval <- as.numeric(as.character(entero_diff_t$pval))
    tmp_qvalue <- p.adjust(tmp_pval_t$pval, method='fdr')
    tmp_pval_adjust <- cbind(tmp_pval_t, tmp_qvalue)
    tmp_pval_adjust05 <- tmp_pval_adjust[tmp_pval_adjust$tmp_qvalue < 0.05,]#tmp_qvalue < 0.0
5,]

    tmp_L6_05 <- L6_rela_fil_sAg_remove_simp[rownames(tmp_pval_adjust05),]
    # tmp_L6_after_05 <- L6_rela_fil_sAg_remove_simp_after[rownames(tmp_pval_adjust05),]
```

```
    tmp_pval_adjust05_n <- cbind(rownames(tmp_pval_adjust05), tmp_pval_adjust05)
    tmp_pval_adjust05_nd <- as.data.frame(tmp_pval_adjust05_n)
    colnames(tmp_pval_adjust05_nd) <- c('genus', 'pval', 'gfc', 'qval')
    tmp_pval_adjust05_nd$genus <- factor(tmp_pval_adjust05_nd$genus, levels = rownames(tmp_pval
_adjust05_nd)[order(sign(tmp_pval_adjust05_nd$gfc)/tmp_pval_adjust05_nd$qval, decreasing = F)])

    min_y = 5

    p1<-ggplot(tmp_pval_adjust05_nd, aes(x = genus))+ #, aes(x = genus), color = sex
      geom_linerange(data = tmp_pval_adjust05_nd, aes(ymin = ifelse(gfc > 0, -min_y, min_y), ym
ax = ifelse(gfc > 0, -min_y-gfc, min_y-gfc), color=ifelse(gfc < 0, "#E7A600", "#4E86C6")), size
= 6, alpha = 0.8)+
      # geom_linerange(data = plot_after_response[plot_after_response$response < plot_after_res
ponse$failure,], aes(ymin = min_y, ymax = ), size = 6, alpha = 0.8, color=)+  -min_y+log10(qval
ue), min_y-log10(qvalue)
      geom_label(aes(x = genus, y = 0, label = genus, family = "sans"),
             inherit.aes = F, fontface = "italic",
             size = 6, label.padding = unit(0.0, "lines"), label.size = 0,
             label.r = unit(0.0, "lines"), fill = "NA", alpha = 0.9, color = "dimgrey")+
      scale_y_continuous(breaks = c(c(-10, -8, -6, -4, -2, 0) - min_y, c(0, 2, 4, 6, 8)+min_y),
                         labels = c("10", "8", "6", "4", "2", "0", "0", "2", "4", "6", "8"))+
      # facet_wrap(~genus, ncol = 2)+
      coord_flip()+
        labs(title="Marked genus in two enterotypes' donors (q<0.05)", x='', y="log2(FoldChang
e)", colour="Cluster")+
      # theme(plot.title = element_text(size=24, hjust = 0.5), axis.title.x = element_text(size
=20, hjust = 0.53),
          # title=element_text(family ="sans", size=21),
            # text=element_text(family ="sans", size=21),aspect.ratio=0.95)+
      theme(text=element_text(family ="sans", size=24), plot.title = element_text(size=26, hjus
t = 0.5), axis.text = element_text(size=24, color ='dimgray'), axis.title.x = element_text(size
=0), axis.title.y = element_text(size=26), axis.ticks = element_blank())+
      theme(legend.position = c(4, .65), legend.background=element_rect(fill = NA), legend.text
= element_text(size=0))+
      scale_colour_manual(values=c("#4E86C6", "#E7A600", "#4D9127", "#90908D", "#7A1D1E", "#C47
737", "#E7A600"))+
      scale_alpha_manual(values = c(0.8))+
      theme(panel.background = element_rect(fill = NA, colour = "lightgrey", size = 3)
            ,axis.line=element_line(colour="lightgrey")
            # ,axis.text.y = element_text(size=0, angle = 0)
            # ,axis.text.x = element_text(size=21, angle = 0)
            ,axis.ticks = element_blank())



    sign_response <- sign(tmp_pval_adjust05$gfc)
    L6_rela_fil_sAg_remove_simp_weighted_don1  <- ifelse(sign_response>0, 1, 0) %*% tmp_L6_05
    L6_rela_fil_sAg_remove_simp_weighted_don2 <- ifelse(sign_response<0, 1, 0) %*% tmp_L6_05

    plot_marked_donor_before <- function(L6_rela_fil_sAg_remove_simp_weighted, flag){
        tmp_3column_before <- meta_fil_config1_na[meta_fil_config1_na$pre_entro %in% c(pre_entr
o) & meta_fil_config1_na$don_entro %in% ifelse(flag, 'donor1', 'donor2'), c('Previous_sra', 'SR
A_Sample', 'Donor_sra', 'postfmt_symptoms')]
        colnames(tmp_3column_before) <- c('Before_r', 'After_r', 'Donor_r', 'post')

        tmp_3column_before$Before <- ((L6_rela_fil_sAg_remove_simp_weighted[,tmp_3column_before
$Before_r]) / mean(L6_rela_fil_sAg_remove_simp_weighted[,tmp_3column_before$Donor_r]))
```

```
        tmp_3column_before$After <- ((L6_rela_fil_sAg_remove_simp_weighted[,tmp_3column_before
$After_r]) / mean(L6_rela_fil_sAg_remove_simp_weighted[,tmp_3column_before$Donor_r]))

        tmp_3column_before$Donor <- ((L6_rela_fil_sAg_remove_simp_weighted[,tmp_3column_before
$Donor_r]) / mean(L6_rela_fil_sAg_remove_simp_weighted[,tmp_3column_before$Donor_r]))

        samples_num <- nrow((tmp_3column_before))

        FMTstage_before <- melt(tmp_3column_before, measure.vars = c('Before', 'After', 'Donor'
)) #

        FMTstage_before$variable <- factor(FMTstage_before$variable, levels = c('Before', 'Afte
r', 'Donor'))
        FMTstage_abun_before <- FMTstage_before
        FMTstage_abun_before$abun <- FMTstage_before$value#tmp_3column_before$After - tmp_3colu
mn_before$Before #FMTstage_before$value
        # FMTstage_abun_before <- cbind(FMTstage_before, L6_rela_fil_sAg_remove_simp_weighted[,
FMTstage_before$value])
        # colnames(FMTstage_abun_before) <- c(colnames(FMTstage_before), 'abun')



        FMTstage_abun_before$abun <- as.numeric(FMTstage_abun_before$abun/samples_num)
        FMTstage_abun_before$post <- factor(FMTstage_abun_before$post, levels = c('response',
'failure'))

        FMTstage_abun_before_1 <- FMTstage_abun_before[(order( -FMTstage_abun_before$abun)),]#F
MTstage_abun_before$post,

        order_1 <- unique(FMTstage_abun_before_1[FMTstage_abun_before_1$variable %in% c('Befor
e'), 'After_r'])
        FMTstage_abun_before_1$After_r <- factor(FMTstage_abun_before_1$After_r, levels = order
_1)
        FMTstage_abun_before_2 <- FMTstage_abun_before_1[order(FMTstage_abun_before_1$After_
r),]

        # FMTstage_abun_before_1$After_r <- factor(FMTstage_abun_before_1$After_r, levels = uni
que())

        library(plyr)


        cumu <- ddply(FMTstage_abun_before_2, .(variable), transform, cumAbun = cumsum(abun))
        # FMTstage_abun_before$abun <- factor(FMTstage_abun_before$abun, levels=unique(FMTstage
_abun_before$abun))
        FMTstage_abun_before_1 <- cumu
        FMTstage_abun_before_1$id <- as.numeric(mapvalues(FMTstage_abun_before_1$After_r, from
 = order_1, to = 1:length(order_1)))/length(order_1)

        FMTstage_abun_before_1
      # FMTstage_abun_before$value <- as.numeric(FMTstage_abun_before$value)


    }

    p2 <- plot_marked_donor_before(L6_rela_fil_sAg_remove_simp_weighted_don1, 1)
    p3 <- plot_marked_donor_before(L6_rela_fil_sAg_remove_simp_weighted_don2, 0)

    p4<-ggplot()+
```

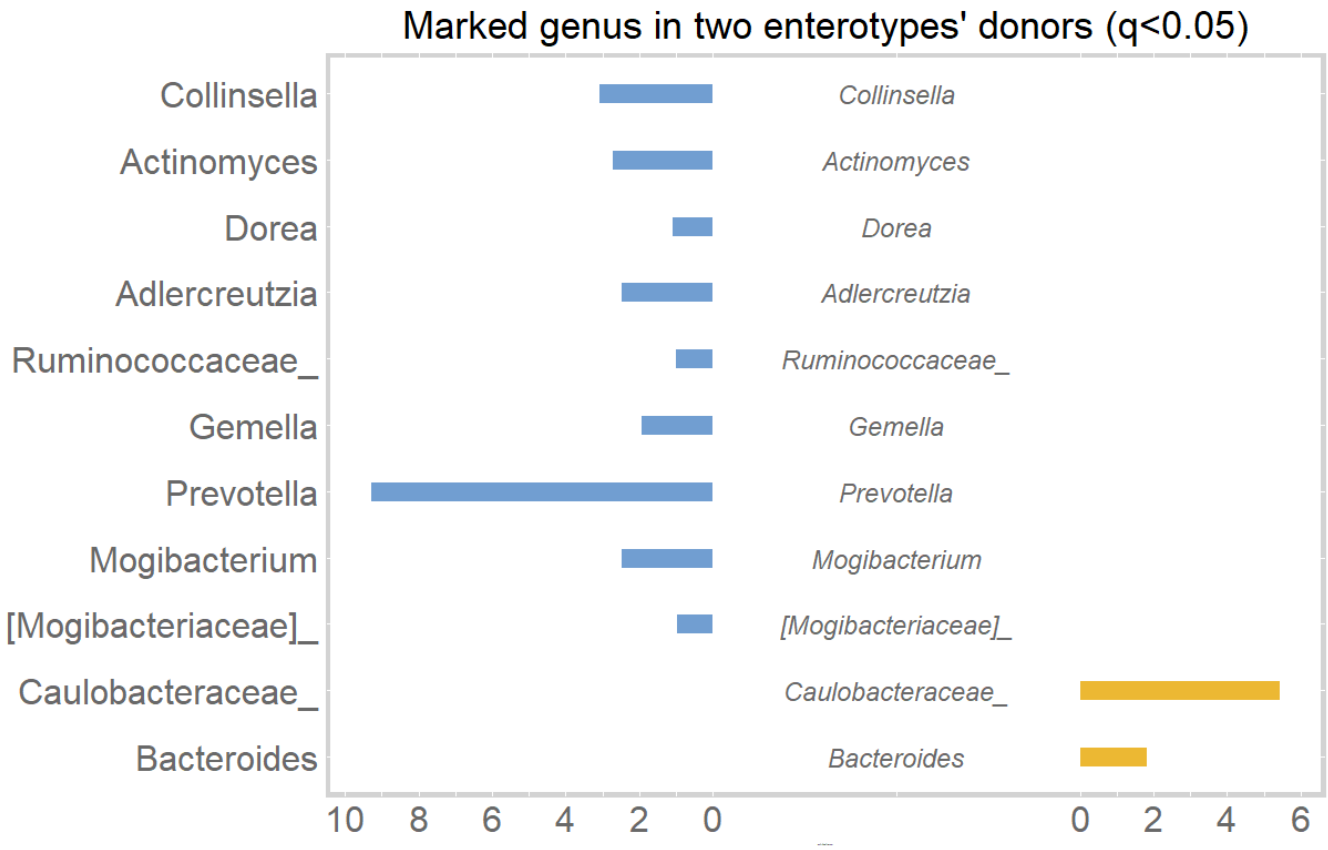
```
      # geom_line(aes(x=id, y=(cumAbun), group=variable, color=variable), p2, linetype = 1, siz
e=1.5)+
      geom_line(aes(x=id, y=cumAbun, group=variable, color=variable), p3, linetype = 2, size=1.
5)+
      scale_color_manual(name="FMT", values=c(donor_before_after_color, "#90908D", 'lightgrey'
))+
        theme(text=element_text(family ="sans", size=32), plot.title = element_text(size=34, hj
ust = 0.5), axis.text = element_text(size=32, color ='dimgray'), axis.title.x = element_text(si
ze=34), axis.title.y = element_text(size=34), axis.ticks = element_line(size=1.5, color ='dimgr
ay'), axis.ticks.length = unit(7, "pt"))+
  theme(aspect.ratio = 1.618, legend.background=element_blank()#, legend.position=c(3.75, 0.6)
        ,panel.background = element_rect(fill = NA, colour = "lightgrey", size = 3)
        ,axis.line=element_line(colour="lightgrey")
        ,legend.key = element_rect(fill = NA, color = NA))+
    guides(colour = guide_legend(override.aes = list(size=5)))+theme(panel.grid.major=element_b
lank(),panel.grid.minor=element_blank());p4

    p5<-ggplot()+
      geom_line(aes(x=id, y=(cumAbun), group=variable, color=variable), p2, linetype = 1, size=
1.5)+
      # geom_line(aes(x=id, y=cumAbun, group=variable, color=variable), p3, linetype = 2, size=
1.5)+
      scale_color_manual(name="FMT", values=c(donor_before_after_color, "#90908D", 'lightgrey'
))+
        theme(text=element_text(family ="sans", size=32), plot.title = element_text(size=34, hj
ust = 0.5), axis.text = element_text(size=32, color ='dimgray'), axis.title.x = element_text(si
ze=34), axis.title.y = element_text(size=34), axis.ticks = element_line(size=1.5, color ='dimgr
ay'), axis.ticks.length = unit(7, "pt"))+
  theme(aspect.ratio = 1.618, legend.background=element_blank()#, legend.position=c(3.75, 0.6)
        ,panel.background = element_rect(fill = NA, colour = "lightgrey", size = 3)
        ,axis.line=element_line(colour="lightgrey")
        ,legend.key = element_rect(fill = NA, color = NA))+
    guides(colour = guide_legend(override.aes = list(size=5)))+theme(panel.grid.major=element_b
lank(),panel.grid.minor=element_blank());p5

    if(dim(p1$data)[1] > 0){list((p1), (p4), (p5))}
}
```

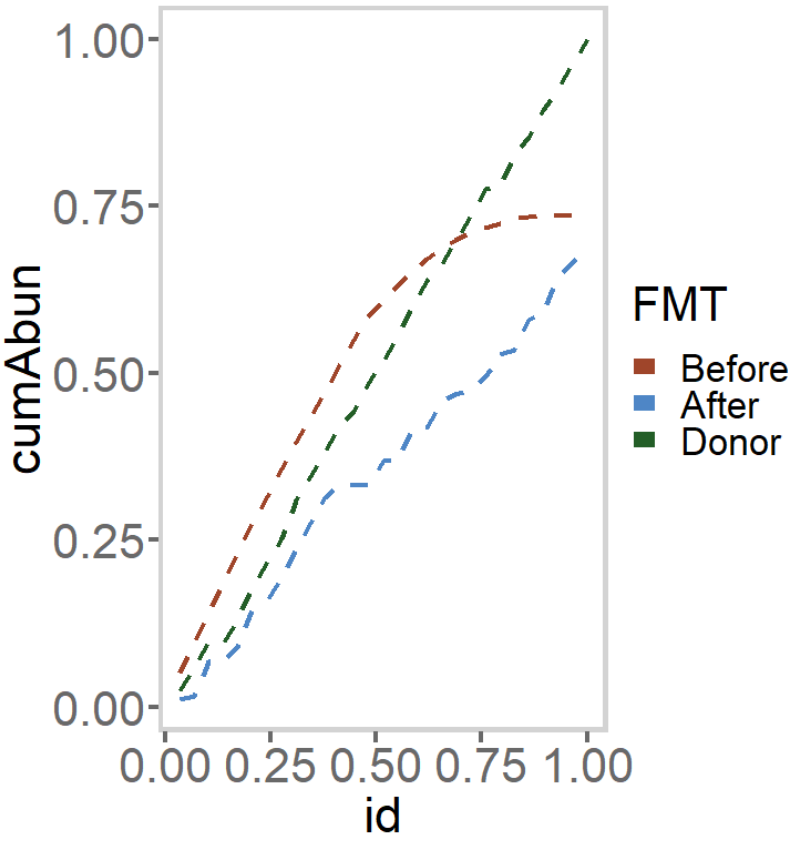
Hide

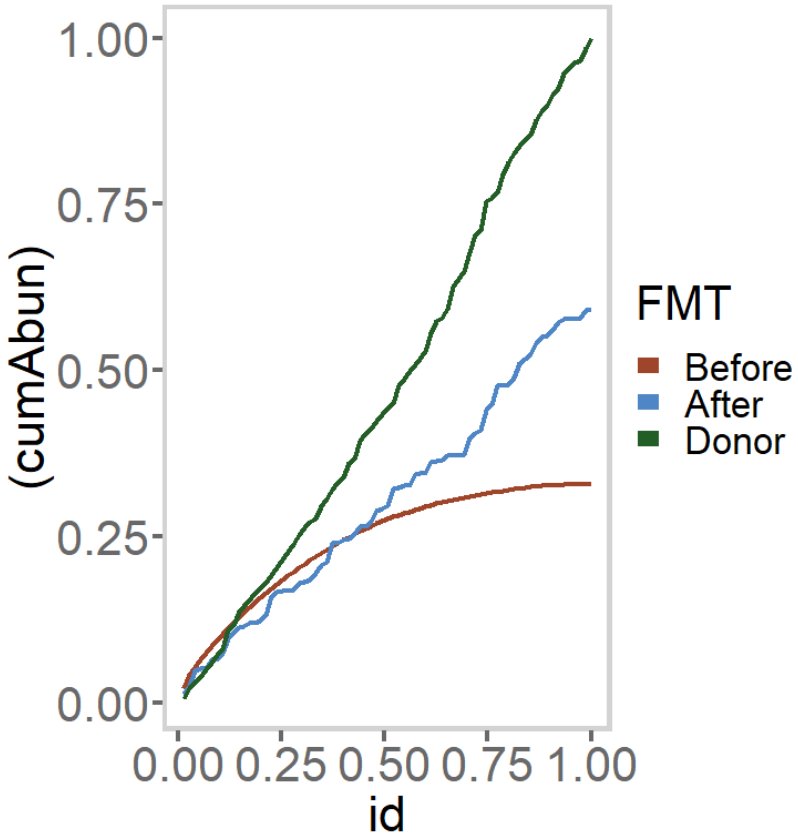
```
# pdf('./figure4//fig4_engrament_donor_marked.pdf', width = 8, height = 8)

donor_engraft('before2')
```

```
The conditional covariance matrix has zero diagonal elementsThe conditional covariance matrix h
as zero diagonal elementsThe conditional covariance matrix has zero diagonal elementsThe condit
ional covariance matrix has zero diagonal elementsThe conditional covariance matrix has zero di
agonal elementsThe conditional covariance matrix has zero diagonal elementsThe conditional cova
riance matrix has zero diagonal elementsThe conditional covariance matrix has zero diagonal ele
mentsThe conditional covariance matrix has zero diagonal elementsThe conditional covariance mat
rix has zero diagonal elementsThe conditional covariance matrix has zero diagonal elementsThe c
onditional covariance matrix has zero diagonal elementsThe conditional covariance matrix has ze
ro diagonal elements
```

```
[[1]]

[[2]]

[[3]]
```


Marked genus in two enterotypes' donors (q<0.05)
Collinsella
Actinomyces
Dorea
Adlercreutzia
Ruminococcaceae_
Gemella
Prevotella
Mogibacterium
[Mogibacteriaceae]_
Caulobacteraceae_
Bacteroides
10 8 6 4 2 0
0 2 4 6


cumAbun
1.00
0.75
0.50
0.25
0.00
0.00 0.25 0.50 0.75 1.00
id
FMT
Before
After
Donor



Hide

```
# dev.off()
```

Hide

```
##enterotype change
table(meta_fil_config1$after_entro, meta_fil_config1$don_entro, meta_fil_config1$postfmt_sympto
ms, meta_fil_config1$pre_entro)
```

```
, ,  = failure,  = before1


        donor1 donor2
  after1      9      2
  after2     16      7
  after3     13      6

, ,  = response,  = before1


        donor1 donor2
  after1     48      7
  after2     17      3
  after3     33     21

, ,  = failure,  = before2


        donor1 donor2
  after1      7      6
  after2      6      5
  after3     10      8

, ,  = response,  = before2


        donor1 donor2
  after1     18      2
  after2     13      1
  after3     21      7
```

Hide

```
# table( meta_fil_config1_na$pre_entro, meta_fil_config1_na$after_entro, meta_fil_config1_na$do
n_entro)
dim(meta_fil_config1_na)
```

```
[1] 286  19
```

Hide

```
library("igraph")
library(markovchain)
```

Hide

```
plot_MCC <- function(transElec, Ecolor){
    mcPreg = new("markovchain", states = c("ET_E", "ET_B", "ET_P"), transitionMatrix = transEle
c)
    # plot(mcPreg, node.size = 10)

    netMC = markovchain:::.getNet(mcPreg, round = TRUE)
    wts = E(netMC)$weight/100
    edgel = get.edgelist(netMC)
    elcat = paste(edgel[,1], edgel[,2])
    elrev = paste(edgel[,2], edgel[,1])
    edge.curved = sapply(elcat, function(x) x %in% elrev)
    default.par = par(no.readonly = TRUE)
    #
    plotMC = function(object, ...) {
      netMC = markovchain:::.getNet(object, round = TRUE)
      plot.igraph(x = netMC, ...)
    }
    #
    vert.sz = 40#*sapply(states(mcPreg), function(x) nrow(unique(sample_data(data_markov)[sampl
e_data(data_markov)$CST==x,"SubjectID"])))

    vert.sz = vert.sz #* 0.85
    vert.font.clrs = c("white", "white", "white", "white", "white")
    #
    edge.loop.angle = c(-1, 1, 0, 0, 3.14, 3.14)-1
    layout = matrix(c(0.4, 0.6, 0.4, 0.4, 0.45, 0.5), ncol=2, byrow=T) #0.6,0.95, 0.43, 1, 0.3,
0.66
    # layout.show(n=3)

    edge.arrow.size=1.5
    edge.arrow.width= 2.5
    edge.width = ifelse(wts==1, 1, (15*wts + 0.5))
    edge.labels = as.character(round((E(netMC)$weight/100), 2))
    edge.labels[edge.labels==.98] = NA  # labels only for self-loops



    plotMC(mcPreg,
           edge.arrow.size=edge.arrow.size, edge.arrow.width = edge.arrow.width,
           edge.label = edge.labels, edge.label.font=1, edge.label.cex=2, edge.label.color='bla
ck',  edge.label.family = 'sans',
           ##
           edge.width=edge.width, edge.curved=edge.curved,edge.color=Ecolor,
           layout=layout, edge.loop.angle = edge.loop.angle,
           vertex.size=(vert.sz),
           vertex.label.font = 1, vertex.label.family = 'sans', vertex.label.cex = 2,
           vertex.label.color = vert.font.clrs, vertex.frame.color = NA, vertex.color = c("#7A1
D1E", "#C47737", '#4E86C6', '#E7A600'))
}
```

Hide

```
# donor enterotype vs after enterotype
#meta_fil_config_entro
# using distance to medoids
# don_data.medoids  pre_data.medoids
after_enter <- (meta_fil_config1[, c('SRA_Sample', 'after_entro')])
medoids <- c(pre_data.medoids[1], don_data.medoids, pre_data.medoids[2])

a_entero <- NULL
for(aj in 1:dim(after_enter)[1]) {
    a = after_enter[aj, ]
    method_dist = 'bray'
    E <- vegdist(rbind(L6_rela_fil_sAg_remove_simp[, as.character(medoids[1])], L6_rela_fil_sAg
_remove_simp[, as.character(a[1])]), method = method_dist)
    P <- vegdist(rbind(L6_rela_fil_sAg_remove_simp[, as.character(medoids[2])], L6_rela_fil_sAg
_remove_simp[, as.character(a[1])]), method = method_dist)
    B <- vegdist(rbind(L6_rela_fil_sAg_remove_simp[, as.character(medoids[3])], L6_rela_fil_sAg
_remove_simp[, as.character(a[1])]), method = method_dist)
    BB <- vegdist(rbind(L6_rela_fil_sAg_remove_simp[, as.character(medoids[4])], L6_rela_fil_sA
g_remove_simp[, as.character(a[1])]), method = method_dist)
    a_entero <- c(a_entero, ifelse(which.min(c(E, P, B)) == 4, 3, which.min(c(E, P, B))))


}
```

Hide

```
#1 E
#2 P
#3 B
entero_like <- cbind(a_entero, meta_fil_config1[, c( 'pre_entro', 'don_entro')])#,  'postfmt_sy
mptoms'
donor1_dat <- table(entero_like)[, , don_entro = 'donor1']

donor2_dat <- table(entero_like)[, , don_entro = 'donor2']
# donor2_dat[3,] = donor2_dat[3,] + donor2_dat[4,]
# donor2_dat[4,] = c(0, 0)
table(entero_like)
```

```
, , don_entro = donor1

        pre_entro
a_entero before1 before2
       1      22      10
       2      98      56
       3      31      20

, , don_entro = donor2

        pre_entro
a_entero before1 before2
       1       8       2
       2      15      17
       3      30      13
```

Hide

```
donor1_tran <- c(t(apply(donor1_dat, 1, function(x){(x / (colSums(donor1_dat)))})))
donor2_tran <- c(t((apply(donor2_dat, 1, function(x){(x / c(colSums(donor2_dat)))}))))
donor1_tran
```

```
[1] 0.1456954 0.6490066 0.2052980 0.1162791 0.6511628 0.2325581
```

Hide

```
donor2_tran
```

```
[1] 0.1509434 0.2830189 0.5660377 0.0625000 0.5312500 0.4062500
```

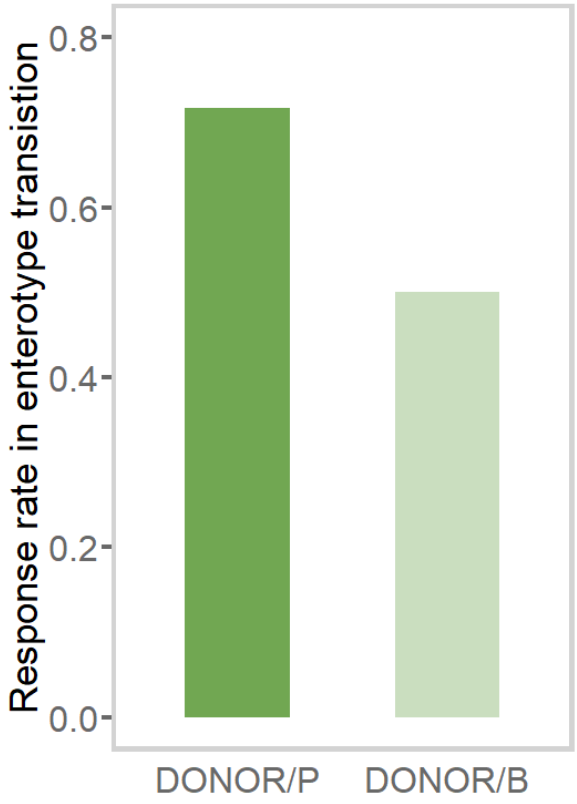
Hide

```
entero_like <- cbind(a_entero, meta_fil_config1[, c( 'don_entro','pre_entro',  'postfmt_symptom
s')])#
table(entero_like)
```

```
, , pre_entro = before1, postfmt_symptoms = failure

        don_entro
a_entero donor1 donor2
       1     13      7
       2     18      2
       3      7      6

, , pre_entro = before2, postfmt_symptoms = failure

        don_entro
a_entero donor1 donor2
       1      2      1
       2     13     12
       3      8      6

, , pre_entro = before1, postfmt_symptoms = response

        don_entro
a_entero donor1 donor2
       1      9      1
       2     71     10
       3     18     20

, , pre_entro = before2, postfmt_symptoms = response

        don_entro
a_entero donor1 donor2
       1      8      1
       2     33      3
       3     11      6
```

Hide

```
plot_change <- data.frame(don_entro=c('DONOR/P', 'DONOR/B'), change=c(33/(33+13), 6/(6+6)))
plot_change$don_entro <- factor(plot_change$don_entro, levels = c('DONOR/P', 'DONOR/B'))
plot_change$change <- as.numeric(plot_change$change)

ggplot(plot_change, aes(x=don_entro, y=change))+#as.factor(change)
  geom_bar(stat="identity",  width = 0.5, fill='#4D9127', alpha=c(0.8, 0.3))+#
    labs(x= c(''), y=c('Response rate in enterotype transistion'), title = c())+
    scale_y_continuous(limits=c(0, 0.8))+
    scale_fill_manual(name="FMT", values=c("#90908D", "#4D9127", "#90908D", "#4D9127", donor_before_after_color, "#962E2B", "#4E86C6", "#4D9127", "#90908D", 'lightgrey'))+
 scale_color_manual(name="FMT", values=c("#90908D", "#4D9127", "#90908D", "#4D9127", donor_before_after_color, 'NA', 'NA', donor_before_after_color, "#962E2B", "#4E86C6", "#4D9127", "#90908D", 'lightgrey'))+#'#C77CFF', '#43AFC8',
   # theme(text=element_text(family ="sans Neue", size=23), plot.title = element_text(size=26, hjust = 0.5), axis.title.x = element_text(size=24, vjust = -0.5, hjust = 0.55, color ='dimgray'), axis.text.x = element_text(size=0))+
  theme(text=element_text(family ="sans", size=24), plot.title = element_text(size=26, hjust = 0.5), axis.text = element_text(size=24, color ='dimgray'), axis.title.x = element_text(size=26), axis.title.y = element_text(size=26), axis.ticks = element_blank())+
 theme(aspect.ratio = 1.62, legend.background=element_blank(), legend.position=c(1.75, 0.6)
       ,panel.background = element_rect(fill = NA, colour = "lightgrey", size = 3)
       ,axis.line=element_line(colour=NA), axis.ticks.y = element_line(size=1.5, color ='dimgray'), axis.ticks.length = unit(7, "pt")
       ,legend.key = element_rect(fill = NA, color = NA), axis.ticks = element_blank(), panel.grid = element_blank())+ #
   guides(colour = guide_legend(override.aes = list(size=5)))
```



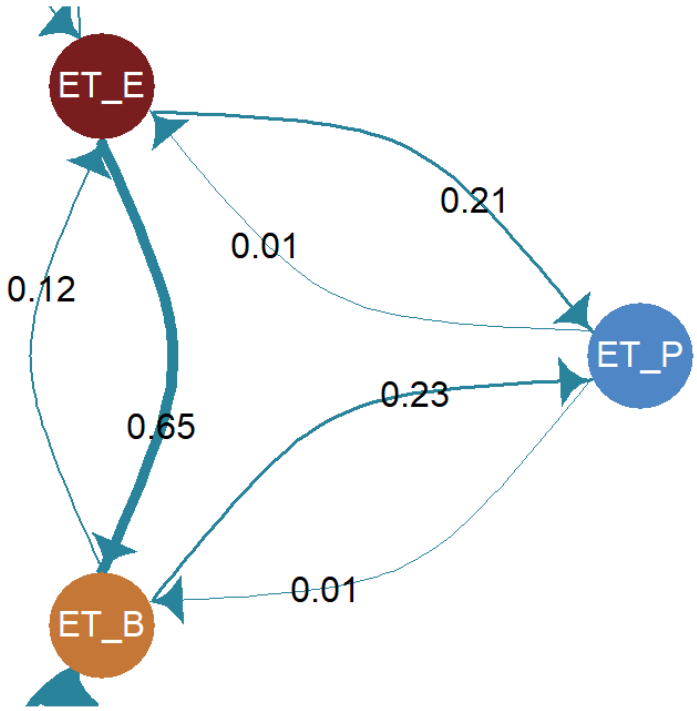
Hide

```
ggsave(paste("./figure4/change_entero", 2, ".pdf", sep = ''), device = "pdf")
```

```
Saving 12.9 x 8 in image
```

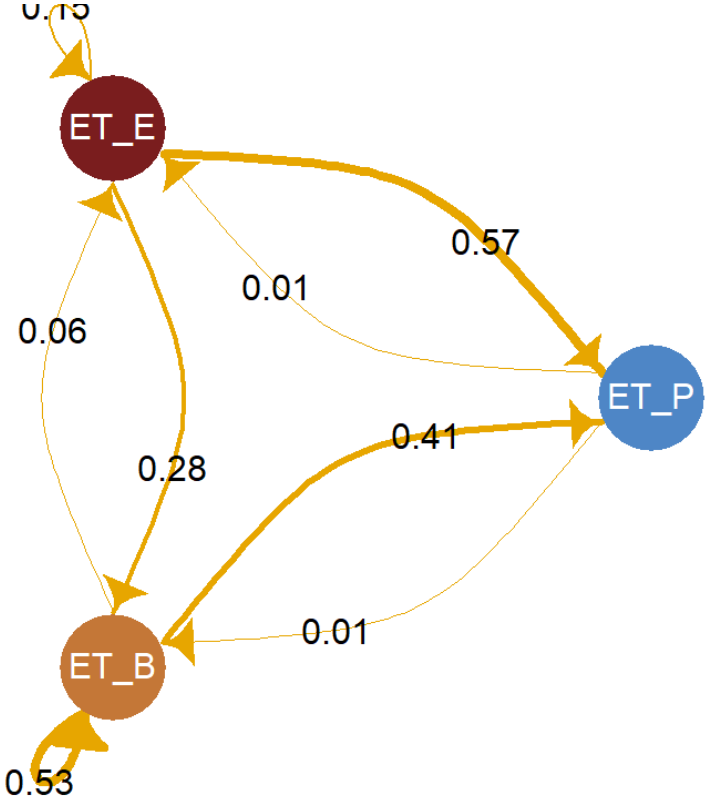
Hide

```
# pdf("figure4/4main_markovchain.pdf", height = 8)
plot_MCC(matrix(c(donor1_tran, 0.01, 0.01, .98), byrow = T, nrow = 3), '#28839B')
```



Hide

```
plot_MCC(matrix(c(donor2_tran, 0.01, 0.01, .980), byrow = T, nrow = 3), '#E7A600')
```



Hide

```
# dev.off()
```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.